Protocolo Híbrido de Ordem Total Uniforme com entrega Optimista

Pedro Vicente
Hugo Miranda
Luís Rodrigues

DI–FCUL TR–01–14

Dezembro 2001

Departamento de Informática
Faculdade de Ciências da Universidade de Lisboa
Campo Grande, 1749–016 Lisboa
Portugal

Technical reports are available at `http://www.di.fc.ul.pt/tech-reports`. The files are stored in PDF, with the report number as filename. Alternatively, reports are available by post from the above address.

Protocolo Híbrido de Ordem Total Uniforme com entrega Optimista

Pedro Vicente Hugo Miranda Luís Rodrigues

Universidade de Lisboa
{pedrofrv,hmiranda,ler}@di.fc.ul.pt

Dezembro 2001

Resumo

Os algoritmos de difusão com ordem total uniforme simplificam o desenvolvimento de aplicações que usam a replicação como técnica para obter tolerância a faltas. Este artigo propõe e compara três alternativas para concretizar um serviço de entrega optimista em protocolos de ordem total uniforme para sistemas de grande escala geográfica.

1 Introdução

A replicação é uma técnica bastante utilizada para obter tolerância a faltas. Uma das abstracções mais relevantes para a concretização da replicação é a difusão fiável com ordem total, por vezes também designada por difusão atómica. Este serviço garante que todas a réplicas correctas recebem exactamente os mesmos pedidos pela mesma ordem, facilitando deste modo a manutenção de um estado mutuamente coerente. Em particular, quando as operações executadas pelas réplicas são deterministas, a coerência mútua é garantida de modo quase automático, concretizando-se uma máquina de estados distribuída [18].

Naturalmente, o desempenho do protocolo que concretiza o serviço de difusão atómica é crucial para o desempenho da aplicação replicada. Por este motivo, vários algoritmos têm sido propostos com a finalidade de optimizar o serviço para diferentes condições operacionais [12, 1, 17]. Antes de referir estes aspectos com maior pormenor, importa salientar que existem duas definições possíveis para um protocolo de ordem total, nomeadamente: a ordem total não-uniforme e a ordem total uniforme. Estas duas primitivas podem ser definidas do seguinte modo:

Ordem Total Não-uniforme *Se dois processos correctos p e q entregam as mensagens m e m', então p entrega m antes de m' se e só se q entrega m antes de m'.*

Ordem Total Uniforme *Se dois processos (correctos ou não) p e q entregam as mensagens m e m', então p entrega m antes de m' se e só se q entrega m antes de m'.*

A diferença entre estas duas definições é substancial, embora subtil numa primeira aproximação. Fundamentalmente, a diferença entre as duas propriedades reside nas

garantias de coerência entre o estado de uma réplica que falha e o estado das réplicas que permanecem correctas. Enquanto a primeira definição permite que uma réplica falhe recebendo algumas mensagens por uma ordem diferente das réplicas que permanecem activas, a segunda definição assegura que todas as mensagens entregues por uma réplica são coerentes, mesmo que esta réplica venha a falhar.

Naturalmente, a segunda definição é mais forte, sendo particularmente útil no caso em que uma réplica pode recuperar após uma falha e manteve parte do seu estado em memória estável. Para além disso, em sistemas assíncronos puros não é possível distinguir um processo falhado de um processo correcto mas lento. Infelizmente, é possível demonstrar que o problema de resolver a ordem total uniforme é equivalente ao problema do consenso distribuído, um problema sem solução em sistemas assíncronos puros [6] e cujas soluções para sistemas aumentados com detectores de falhas não fiáveis requerem diversos passos de comunicação [3, 7].

Apesar da complexidade inerente aos protocolos de ordem total uniforme, o recurso à difusão atómica como técnica para gerir a replicação mostrar-se vantajosa, não só por permitir soluções elegantes e modulares, mas também por permitir obter ganhos de desempenho, como demonstram os trabalhos recentes na área de replicação em bases de dados [13, 11].

Uma das técnicas para obter melhor desempenho em sistemas baseados em difusão atómica uniforme consiste na utilização de protocolos *optimistas*. Estes protocolos baseiam-se no facto de que a ordem pela qual as mensagem vão ser ordenadas pode ser geralmente estimada antes da terminação do protocolo: em execuções em que não ocorrem falhas de nós ou partições na rede (as mais frequentes) esta ordem estimada é tipicamente a ordem estabelecida pelo algoritmo. Deste modo, os protocolos optimistas informam previamente a aplicação de qual a ordem prevista para as mensagem, permitindo a execução de algum pré-processamento antes da confirmação definitiva desta *ordem estimada*. Esta técnica é particularmente útil em sistemas transaccionais, tais como os utilizados nas bases de dados, uma vez que uma execução optimista de uma transacção pode ser abortada (e eventualmente re-submetida) caso a ordem definitiva venha a ser diferente da ordem estimada.

Este artigo aborda o problema de concretizar protocolos de ordem total uniforme optimistas para redes de grande escala geográfica (este tipo de redes possui características próprias como, por exemplo, uma grande variância nos atrasos das mensagens). Para isso, propõe diferentes alternativas de combinar e extender protocolos de ordem total (não-uniforme) optimizados para redes de grande escala com primitivas que assegurem a uniformidade da ordenação final. O desenvolvimento de protocolos com estas características é extremamente útil para a concretização de bases de dados replicadas sobre larga escala geográfica.

O artigo encontra-se estruturado do seguinte modo. Na Secção 2 é revisto o modelo de sistema e o trabalho relacionado. Posteriormente, a Secção 3 apresenta três alternativas distintas para a concretização de um protocolo de ordem total uniforme optimista adequado a redes de grande escala, os quais são comparados na Secção 4. Finalmente a Secção 5 conclui o artigo.

2 Modelo de Sistema e Trabalho relacionado

Esta secção descreve com um pouco mais de pormenor o modelo de sistema assumido e posteriormente faz uma breve panorâmica sobre os principais trabalhos onde assenta o desenvolvimento dos novos algoritmos.

2.1 Sistemas assíncronos

Como foi referido o objectivo deste trabalho é encontrar novos algoritmos de ordenação total optimista adequados ao funcionamento em sistemas com dispersão geográfica. Estes sistemas são caracterizados pela existência de uma grande variância nos atrasos das mensagens, pelo que são modelados apropriadamente por um sistema assíncrono. Ou seja, nos algoritmos discutidos neste artigo não se assume a existência de um limite máximo para o tempo de entrega ou processamento de mensagens por parte da rede e dos nós.

Com foi também já mencionado, o problema do acordo distribuído não possui solução em sistemas assíncronos puros [6]. Este resultado de impossibilidade pode ser contornado através da introdução de detectores de falha não fiáveis [3], cuja disponibilidade é assumida. Para além disso, e para garantir o progresso do algoritmo, é necessário assumir que apenas uma minoria dos processos pode falhar.

2.2 Arquitectura de software

O desenvolvimento do protocolo de ordem total optimista é facilitado através de um recurso a arquitecturas modulares que encorajem a re-utilização e composição de serviços mais básicos. Esta é a aproximação seguida por este trabalho. Deste modo, assume-se a existência de um conjunto de serviços auxiliares que, dependendo do algoritmo, poderão ser combinados de formas distintas. Os serviços que se assumem disponíveis são os seguintes:

- Serviço de difusão em grupo não fiável.
- Serviço de filiação particionável associado à difusão em grupo fiável não-uniforme oferecendo uma semântica de *sincronia virtual* [2]. A sincronia virtual pode ser definida informalmente do seguinte modo:

  Sincronia Virtual *A evolução dos membros do grupo é representada pela entrega de vistas com a filiação corrente do grupo. A sincronia virtual garante que se uma mensagem m é entregue numa vista V^i então todos os processos correctos p que pertencem a V^i também entregam m nessa vista.*

- Serviço de filiação e difusão em grupo fiável uniforme.
- Serviço de ordenação total não-uniforme adaptado a redes de grande escala.

De notar que a maioria destes serviços se encontram concretizados, tendo alguns deles sido concebidos e/ou desenvolvidos pelos autores deste artigo. A única excepção é o serviço de filiação e difusão em grupo fiável uniforme que não se encontra ainda disponível.

Para facilitar a concretização do protocolo, recorre-se a uma plataforma de suporte ao desenvolvimento, composição e execução de protocolos de comunicação designada por *Appia* [14]. O *Appia* foi desenvolvido em Java e toma partido das características da programação orientada a objectos. A grande vantagem do *Appia* em relação a outros sistemas similares [9, 10] é permitir criar pilhas de protocolos mais flexíveis com estruturas complexas. De notar que os componentes listados acima se encontram concretizados no sistema *Appia*.

2.3 Protocolos Optimistas

A ideia chave dos protocolos optimistas consiste em realizar antecipadamente algum processamento, antes de conhecido o resultado final de um algoritmo de ordenação, com base numa estimativa desse resultado. Este princípio tem sido utilizado extensivamente no controlo de concorrência em bases de dados e foi recentemente estendido aos protocolos de comunicação para suporte ao processamento transaccional. Os protocolos optimistas informam a aplicação assim que obtêm uma estimativa para a ordem final, confirmando ou cancelando esta estimativa mais tarde, através da entrega de uma ordem definitiva. Salientamos dois protocolos com estas características.

Difusão Atómica Optimista [15] baseia-se no protocolo de Chandra-Toueg [3] que explora o facto de, numa rede local baseada em meio partilhado, as mensagens serem ordenadas de forma total ao atravessarem a rede. Deste modo, a ordem pela qual as mensagens são recebidas da rede é usada como estimativa. Posteriormente, os restantes passos do algoritmo de Chandra-Toueg são executados para assegurar a uniformidade da decisão.

Em [5] é apresentado um mecanismo de replicação activa optimista, que usa um protocolo optimista para ordenar totalmente as mensagens. O protocolo utilizado é baseado num sequenciador, isto é, um dos membros do grupo fica responsável por ordenar as mensagens. A ordenação proposta pelo sequenciador é utilizada como estimativa da ordem definitiva uma vez que o protocolo assegura que, na ausência da detecção de falhas, esta é a ordem estabelecida de modo uniforme. Na ocorrência de uma detecção de falha é utilizado um protocolo de consenso para garantir que as mensagens são ordenadas uniformemente. Se a ordem definitiva diferir da ordem estimada, as alterações efectuadas às réplicas são desfeitas e re-executadas pela ordem definitiva. Para evitar que os clientes recebam resultados incoerentes o sistema de comunicação do cliente só entrega mensagens depois de receber uma maioria de respostas iguais (assegurando a uniformidade do resultado).

2.4 Protocolo de Ordem Total Híbrida

Nos protocolos anteriores, as técnicas utilizadas para obter a ordem estimada não são adequadas às redes de grande escala geográfica, ou porque se baseiam em características exclusivas da redes locais, ou porque se baseiam num único sequenciador.

Como ponto de partida para o desenvolvimento de um protocolo optimista para redes de grande escala foi escolhido o protocolo de ordem total híbrida (não uniforme) apresentado em [17], dado que este foi especialmente concebido para este tipo de redes. Este protocolo utiliza uma combinação entre os protocolos de ordem total baseada em sequenciador [12] e os protocolos de ordem total baseada em relógios lógicos [16, 4].

A ordem total baseada em sequenciador adapta-se melhor a redes locais em que os pedidos são raros, enquanto a ordem baseada em relógio lógicos se adapta melhor a grande escala e quando todos os nós estão a enviar mensagens. O protocolo usa sequenciador para ordenar as mensagens dos nós mais próximos e mais lentos (em que o número de mensagens enviadas é menor) e usa relógios lógicos para ordenar as mensagens dos nós mais distantes.

O protocolo híbrido na sua versão original apenas garante ordem total não uniforme. Isto porque as mensagens ordenadas pelo sequenciador podem ser entregues ao próprio antes deste falhar sem conseguir propagar a ordem de entrega para os restantes elementos do grupo. Para além disso, dado que o protocolo não assume uma camada de difusão fiável uniforme, quando se verificam partições na rede, o algoritmo

de ordenação por relógios lógicos pode entregar mensagens diferentes em partições diferentes.

Neste protocolo os membros do grupo podem assumir um papel activo (processos que ordenam as suas mensagens) ou passivo (processos que delegam noutro processo a tarefa de ordenar as suas mensagens). Podem existir vários processos activos e passivos no grupo. Os processos activos atribuem às mensagens uma estampilha lógica que é usada no processo de ordenação. Os processos passivos enviam as suas mensagens indicando qual dos processos activos fica responsável pela ordenação das mesmas. A ordem final das mensagens é estabelecida ordenando de forma total todas as estampilhas geradas pelos processos activos (cada processo tem de esperar até ter mensagens de todos os processos activos antes de poder entregar uma mensagem).

Para optimizar a entrega de mensagens usando relógios lógicos utiliza-se uma técnica designada por sincronização de ritmo[1]. Com esta optimização os processos mais lentos em vez de avançarem o seu relógio em uma unidade, avançam ao ritmo do processo mais rápido.

3 Algoritmos optimistas para grande escala

Nesta secção propõem-se três alternativas distintas para incorporar as ideias do protocolo híbrido descrito anteriormente de modo a obter um protocolo de ordem total uniforme optimista adequado a redes de grande escala geográfica. Cada uma destas alternativas será descrita de seguida.

3.1 Execução sequencial do Algoritmo Híbrido e de Consenso

Esta alternativa consiste em utilizar numa primeira fase o algoritmo híbrido de ordem total [17] para obter de forma eficiente uma estimativa e, posteriormente, usar este valor estimativo como valor proposto para um protocolo de consenso uniforme como, por exemplo, o protocolo de Chandra-Toueg [3].

Note-se que na ausência de falhas ou falsas detecções de falha, o algoritmo híbrido irá propor a mesma ordem a todos os nós. Uma vez que todos os processos irão propor exactamente o mesmo valor, este será também o valor final acordado pelo protocolo.

3.2 Protocolo Híbrido Uniforme

Esta alternativa consiste em estender o protocolo híbrido de ordem total [17] de modo a que este passe a assegurar a uniformidade. Esta extensão consiste em acrescentar trocas de mensagens adicionais ao protocolo de modo a assegurar que uma determinada ordenação é conhecida por uma maioria de processos antes da ser considerada definitiva. Tal como no ponto anterior o protocolo híbrido de ordem total seria usado mas desta vez de uma maneira integrada de modo a garantir mais eficiência.

Tal como anteriormente, o protocolo básico seria usada para obter a estimativa mas agora, em vez de executar um algoritmo de consenso de forma modular, este seria integrado no protocolo para uma maior eficiência.

O protocolo original seria então estendido do seguinte modo. Assim que um nó obtém a sua estimativa (através do protocolo não uniforme), envia esta estimativa para os restantes elementos do grupo. Quando a mesma estimativa é confirmada por uma maioria de processos no sistema, a ordem é considerada definitiva e a aplicação é notificada

[1] Do Inglês, *rate-synchronization*.

que a ordem tentativa se tornou uniforme. A sobrecarga de mensagens introduzida por este passo pode ser evitada se as estimativas forem agregadas à próxima mensagem de dados a enviar. Neste caso, não é necessário trocar mensagens adicionais mas existe uma maior latência na confirmação da ordem uniforme das mensagens para as quais já foi entregue uma estimativa. Outra alternativa para diminuir o número de mensagens trocadas é centralizar o protocolo, isto é, quando os nós obtém a estimativa enviam uma confirmação, numa mensagem ponto-a-ponto, a um membro especifico do grupo. Esse coordenador ao receber uma maioria de confirmações envia uma confirmação final para o grupo indicando que a mensagem foi recebida por uma maioria e logo pode ser entregue.

Para lidar com os casos de falha e partições, o protocolo recorre aos serviços de filiação e difusão em grupo assegurados por uma pilha oferecendo sincronia virtual [2]. Este serviço garante a entrega de uma nova vista quando uma falha é detectada (encapsulando os detectores de falha não fiáveis, que não são acedidos directamente pelo algoritmo de ordem total). Note-se que não se assume a disponibilidade de um serviço de sincronia virtual garantindo entrega uniforme de mensagens ou vistas. Assim, cada processo pode receber sequências de vistas diferentes. Para além disso, quando é recebida uma vista, podem subsistir mensagens para as quais não foi estabelecida nenhuma estimativa e mensagens para as quais não foi estabelecida a ordem uniforme. Como tal, após a recepção de uma nova vista, cada processo p deve executar os seguintes passos:

1. p verifica se a nova vista é maioritária. Apenas os nós que estão numa vista maioritária prosseguem o algoritmo. Os restantes processos deixam de poder processar actualizações e devem executar um procedimento de re-integração no grupo quando voltarem a ter conectividade (este procedimento é executado pelos níveis protocolares superiores).

2. p difunde para os restantes elementos da sua vista uma mensagem indicando que está pronto para ordenar de forma determinista as mensagens pendentes.

3. Quando p recebe uma maioria de mensagens de confirmação das vistas pode entregar as mensagens que ficaram pendentes. p pode ter dois tipos de mensagens por entregar.

   - Mensagens que ainda não têm ordem.
   - Mensagens com ordem estimada, mas que nunca receberam um número suficiente de confirmações.

   Estas mensagens podem ser entregues por um qualquer critério determinista[2] desde que as mensagens com ordem sejam entregues primeiro.

4. A nova vista é entregue à aplicação e o protocolo reinicia a sua execução normal.

3.3 Protocolo Híbrido sobre Difusão Uniforme Optimista

Esta alternativa passa por executar o protocolo híbrido sobre uma camada de difusão uniforme de mensagens, sem realizar alterações significativas ao modo de funcionamento do algoritmo. Note-se no entanto que nem todas as mensagens trocadas pelo protocolo híbrido necessitam de qualidade de serviço uniforme. Em particular, para

[2]Usando um algoritmo que garanta a mesma ordem em todos os nós.

um melhor desempenho, interessa que algumas mensagens sejam transmitidas usando uma primitiva não-uniforme (mais eficiente).

Apesar de esta alternativa parecer relativamente simples de concretizar, para ser eficaz requer um serviço de difusão uniforme optimista. Para compreender esta necessidade é necessário descrever como é concretizado um protocolo de difusão uniforme.

Tal como no caso da ordem total, um protocolo de difusão uniforme necessita de obter a confirmação de que a mensagem a entregar já é conhecida de uma maioria dos nós (e pode, deste modo, ser sempre recuperada mesmo que uma minoria de processos falhe). Para obter este conhecimento, os processos trocam entre si periodicamente informação acerca de quais as mensagens que já receberam. Quando uma mensagem já é conhecida por uma maioria de processos, esta diz-se *estável*. Por este motivo, os protocolos que mantêm esta informação são também designados por protocolos de manutenção de estabilidade [8]. Deve-se salientar que este tipo de protocolos permite também incorporar mecanismos de detecção de omissões e estimular o pedido de retransmissão de mensagens perdidas.

Como se pode ver, a sobreposição do protocolo híbrido sobre um protocolo de difusão uniforme inviabiliza uma entrega optimista eficiente uma vez que todas as mensagens são atrasadas antes de serem entregues ao protocolo de ordem total. A única forma de contornar este problema consiste em desenvolver um protocolo de difusão fiável uniforme também com entrega optimista. Isto é, o protocolo de difusão uniforme deve também entregar mensagens ao protocolo de ordem total antes de assegurar a uniformidade da entrega. O protocolo de ordem total deve estabelecer uma ordem para esta mensagem no pressuposto que a sua entrega uniforme será confirmada. Posteriormente, caso se verifique impossível assegurar a uniformidade da entrega, este número de ordem deve ser cancelado.

4 Comparação e discussão

Esta secção compara as três alternativas apresentadas anteriormente. Dois aspectos foram considerados nesta comparação: número de passos de comunicação e latência.

4.1 Número de passos de comunicação

O número de passos de comunicação reflecte o número de mensagens que tem de ser enviado sequencialmente pelo protocolo. Por outras palavras, o número de passos indica que que maneira a latência da rede influencia o protocolo. Por exemplo, num protocolo que exija o envio de uma mensagem e a recepção de uma confirmação existe um factor de dois a multiplicar pela latência da rede associado ao tempo de terminação do protocolo.

Note-se que a latência do protocolo pode ser maior caso se tente minimizar o número de mensagens agregando passos de execuções diferentes nas mesmas mensagens (*piggybacking*). Este aspecto será discutido mais adiante.

Começamos esta análise por discutir o número de passos que cada componente requer quando executado isoladamente:

- O protocolo de ordenação total híbrida não uniforme requer no mínimo 1 ou 2 passos de comunicação, consoante o emissor se encontra no modo activo (envia o número de ordem na própria mensagem) ou no modo passivo (um passo para enviar a mensagem e outro para enviar o número de ordem da mensagem), respectivamente.

- O número de passos do protocolo de consenso depende do tipo de algoritmo que é concretizado. Um algoritmo coordenado necessita de pelo menos 3 passos, enquanto que um algoritmo descentralizado (em que todos os processos enviam mensagens para todos, e portanto, mais dispendioso em termos de número de mensagens) necessita geralmente de 2 passos (neste caso, é possível executar o algoritmo em apenas 1 passo, em execuções em que não há falhas e em que todos os processos propõem o mesmo valor).

- A difusão fiável uniforme requer pelo menos 1 passo para executar uma difusão fiável (não uniforme) e requer logicamente mais um passo para garantir que a mensagem foi vista por uma maioria de nós.

Deste modo a primeira alternativa (execução sequencial do protocolo híbrido e do consenso) requer no mínimo 2 passos (nó activo) ou 3 passos (nó passivo), embora numa concretização menos exigente em termos de recursos, recorrente a um algoritmo coordenado para o consenso, demore entre 4 e 5 passos de comunicação.

A alternativa de estender o protocolo híbrido acrescenta ao protocolo não uniforme 1 passo (versão descentralizada) ou 2 passos (versão coordenada) para um mínimo de 2 passos (processo activo, protocolo de controlo descentralizado) e um máximo de 4 passos (processo passivo, protocolo de controlo centralizado).

Na última alternativa (protocolo híbrido sobre difusão uniforme), e utilizando uma entrega optimista ao nível os dois protocolos podem ser executados em paralelo, pelo que o número de passos é limitado pelo tempo necessário para executar a difusão uniforme (entre 2 e 3, consoante os cenários), sendo um valor típico 2 passos de comunicação.

Das três soluções apresentadas a última é a que necessita menos passos para assegurar a entrega de mensagens. A segunda solução também oferece uma solução eficiente em termos de número de mensagens não precisando de se alterar o sistema de comunicação em grupo. A primeira solução é em média mais pesada que as outras.

4.2 Latência

A latência é directamente influenciada pelo número de passos do algoritmo uma vez que em sistemas de grande escala o atraso na rede é frequentemente o factor determinante. No entanto é necessário ter em consideração os seguintes aspectos:

- Sempre que se usa a agregação de mensagens para diminuir a carga da rede a latência aumenta, uma vez que um dado passo de uma instância do algoritmo é atrasado até ao inicio de um passo de outra instância.

- O uso de protocolos descentralizados obriga todos os nós a receberem muito mais mensagens e, caso o suporte para difusão não esteja disponível ao nível da rede (IP multicast), obriga também à troca de um número significativamente maior de mensagens. Dado que a capacidade de processamento dos nós e o débito da rede é limitado, esta sobrecarga pode afectar negativamente a latência.

- Finalmente, a latência média do algoritmo híbrido depende da taxa de transmissão dos diversos nós em relação aos atrasos da rede.

- Os protocolos de estabilidade associados à difusão fiável uniforme, usam geralmente trocas periódicas de mensagens, não respondendo instantaneamente à recepção de uma mensagem.

Atendendo a estes factores, é bastante difícil comparar a latência de cada uma das soluções sem ser experimentalmente.

5 Conclusões e trabalho futuro

Neste artigo foram apresentadas e discutidas várias alternativas à concretização de um algoritmo de ordem total uniforme com entrega optimista. Na análise realizada conseguiu-se compreender parcialmente as vantagens e desvantagens das várias alternativas e qual o trabalho futuro a seguir.

Das três alternativas apresentadas, a que estende o protocolo híbrido original e a que implementa um sistema de comunicação em grupo com uniformidade parecem ser as que fornecem melhor desempenho.

Pode-se afirmar que a solução que fornece uniformidade ao nível do sistema de comunicação em grupo permite uma maior flexibilidade do sistema, permitindo tirar partido das novas funcionalidades em protocolos futuros. Por outro lado, estender o protocolo híbrido original é uma solução que permite mais optimizações ao nível do protocolo.

O trabalho futuro passa por concretizar as várias alternativas sobre a plataforma *Appia* e realizar testes de desempenho sobre essas concretizações de modo a aferir qual das soluções é a mais eficiente.

Referências

[1] Y. Amir, L. Moser, P. Melliar-Smith, D. Agarwal, and P. Ciarfella. Fast message ordering and membership using a logical token-passing ring. In *Proceedings of the 13th International Conference on Distributed Computing Systems*, pages 551–560, Pittsburgh, Pennsylvania, USA, May 1993.

[2] K. Birman and T. Joseph. Reliable Communication in the Presence of Failures. *ACM, Transactions on Computer Systems*, 5(1), February 1987.

[3] T. Chandra and S. Toueg. Unreliable failure detectors for reliable distributed systems. *Journal of the ACM*, 43(2):225–267, 1996.

[4] D. Dolev, S. Kramer, and D. Malki. Early delivery totally ordered multicast in asynchronous environments. In *Digest of Papers, The 23th International Symposium on Fault-Tolerant Computing*, pages 544–553, Toulouse, France, June 1993. IEEE.

[5] Pascal Felber and André Schiper. Optimistic active replication. In *Proceedings of 21st International Conference on Distributed Computing Systems (ICDCS'2001)*, Phoenix, Arizona, USA, April 2001. IEEE Computer Society.

[6] M. Fischer, N. Lynch, and M. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the Association for Computing Machinery*, 32(2):374–382, April 1985.

[7] R. Guerraoui, M. Hurfin, A. Mostefaoui, R. Oliveira, M. Raynal, and A. Schiper. Consensus in asynchronous distributed systems: a concise guided tour. In S. Krakowiak and S. Shrivastava, editors, *Advances in Distributed Systems*, LNCS 1752, chapter 1, pages 33–47. Springer Verlag, 2000.

[8] K. Guo, W. Vogels, and R. Renesse. Structured virtual synchrony: Exploiting the bounds of virtually synchronous group communication. In *Proceedings of the 7th ACM SIGOPS European Workshop*, Connemara, Ireland, May 1996.

[9] M. Hayden. *The Ensemble System*. PhD thesis, Cornell University, Computer Science Department, 1998.

[10] M. Hiltunen, R. Schlichting, and G. Wong. Implementing integrated fine-grain customizable qos using cactus. In *Fast Abstracts, The 29th International Symposium on Fault-Tolerant Computing Systems*, pages 59–60, Madison, Wisconsin, USA, June 1999.

[11] JoAnne Holliday, Divyakant Agrawal, and Ami El Abbadi. Using multicast communication to reduce deadlock in replicated databases. In *actas do IEEE Symposium on Reliable Distributed Systems (SRDS2000)*, October 2000.

[12] M.Frans Kaashoek, Andrew S. Tanenbaum, Susan Flynn Hummel, and Henri E. Bal. An efficient reliable broadcast protocol. *Operating Systems Review*, 23:5–19, October 1989.

[13] B. Kemme and G. Alonso. A suite of database replication protocols based on group communication primitives. In *actas da 18th International Conference on Distributed Computing Systems (ICDCS)*, Amsterdam, Holanda, May 1998.

[14] H. Miranda, A. Pinto, and L. Rodrigues. Appia, a flexible protocol kernel supporting multiple coordinated channels. In *Proceedings of the 21st International Conference on Distributed Computing Systems*, pages 707–710, Phoenix, Arizona, April 2001. IEEE.

[15] F. Pedone and A. Schiper. Optimistic atomic broadcast. In *Proceedings of the 12th International Symposium on Distributed Computing (DISC'98)*, 1998.

[16] L. Peterson, N. Buchholz, and R. Schlichting. Preserving and using context information in interprocess communication. *ACM Transactions on Computer Systems*, 7(3):217–146, August 1989.

[17] L. Rodrigues, H. Fonseca, and P. Veríssimo. Totally ordered multicast in large-scale systems. In *Proceedings of the 16th International Conference on Distributed Computing Systems*, pages 503–510, Hong Kong, May 1996. IEEE.

[18] F. Schneider. Implementing fault-tolerant services using the state machine approach: a tutorial. *ACM Computing Surveys*, 22(4):290–319, December 1990.